Automatic Pet Feeder

COMPILADORES Y DESARROLLO DE LIBRERIAS

LOPEZ ALVAREZ, SANDRA

Introducción

En la actualidad el término de Internet de las Cosas aparece con tanta frecuencia y ha sido tema de investigación en diferentes contextos tales como seguridad, hogar, sistemas de gestión automática, etc.
Internet de las Cosas es la conexión (a través de Internet) de objetos del mundo físico que son equipados con sensores, actuadores y tecnología de comunicación.

En este proyecto, las tecnologías de IoT se enfocan en la necesidad de estar al cuidado que se debe de tener a las macotas sin tener que estar presente en casa y cuidar a la mascota para que tenga una buena calidad de vida.

Descripción del Proyecto

Implementación de las tecnologías de Internet of Things (IoT) para un sistema de gestión inteligente para el cuidado de las mascotas, en específico la alimentación.

Al llevar a cabo este Proyecto, se pretende atender una de las necesidades básicas para el cuidado de los perros y facilitar al dueño la realización de estas actividades como la alimentación.

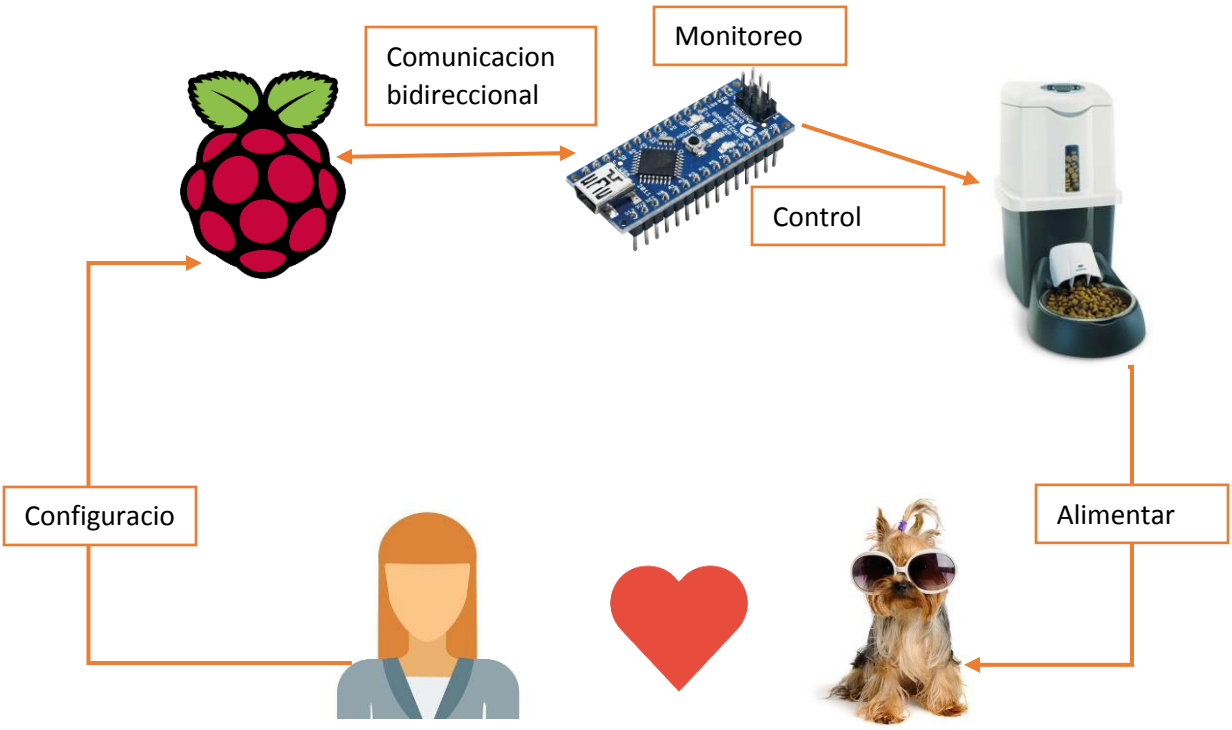
Se utilizara una Raspberry Pi 3 que actuara como servidor para que el usuario introduzca la configuración deseada para que la Raspberry lleve el control de la alimentación del (los) perrito(s).

Se utilizara también un arduino nano, el cual será el que controlara el funcionamiento del despachador de alimento.

Entre el arduino y la Raspberry deberá haber comunicación bidireccional ya que la Raspberry tiene la configuración general como porción (chica, mediana, grande) horarios de alimentación diaria y cuando se cumpla la hora de alimentar le mande la señal al arduino cuando se deba alimentar al perrito y la porción que se debe servir.

El arduino recibe la señal para alimentar y la porción que se deberá servir, entonces el arduino se encarga de controlar el proceso de alimentación dependiendo la porción deseada.

Además el arduino deberá estar monitoreando cuanto alimento hay en el despachador, para cuando quede poca comida el arduino envíe una señal a la Raspberry y esta se encargue de enviar una push notification para avisarle al dueño que hay poca comida y necesita poner más alimento en el despachador.

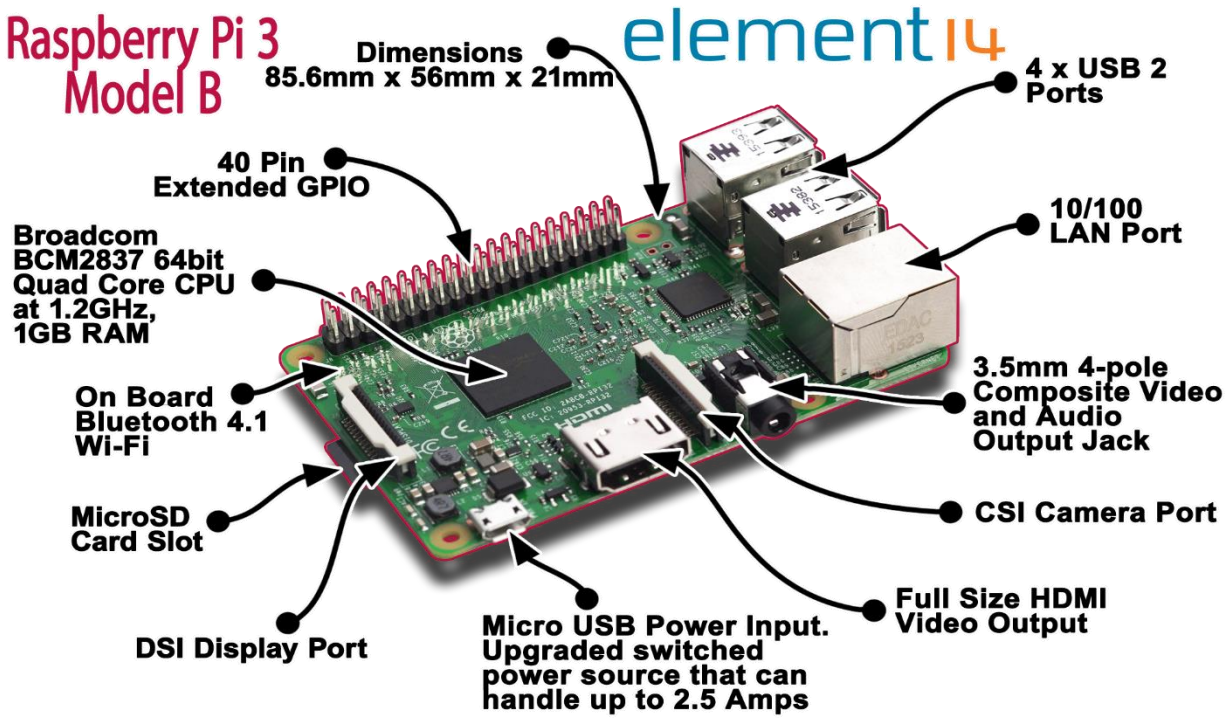


Raspberry Pi 3

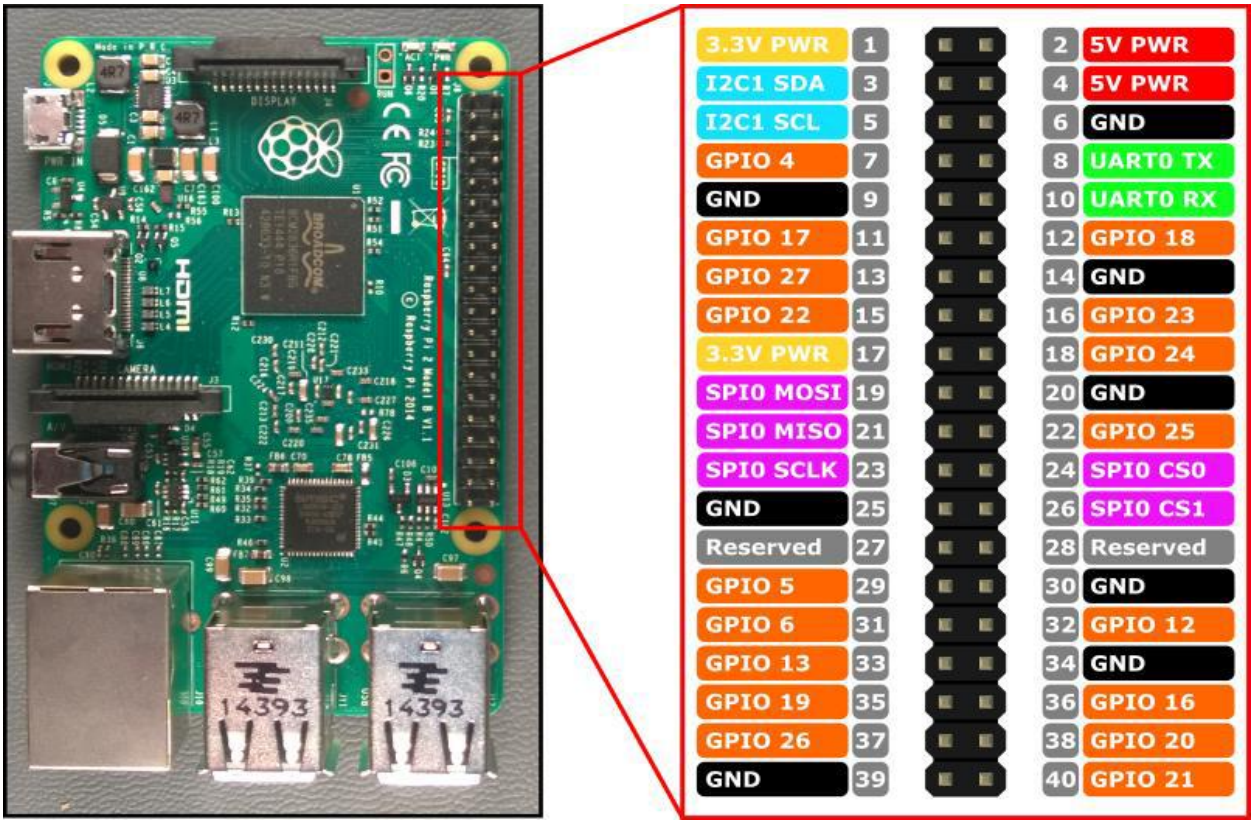
Raspberry Pi es un computador de placa reducida, computador de placa única o computador de placa simple (SBC) de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El software que utiliza es open source, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspbian, aunque permite otros sistemas operativos, incluido una versión de Windows 10. El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos "Turbo" para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía),10 un procesador gráfico (GPU) VideoCore IV, y 512 MB de memoria RAM (aunque originalmente al ser lanzado eran 256 MB). El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente.

Dependiendo el modelo de la Raspberry son las especificaciones con las que cuenta, para este proyecto se utilizó la Raspberry Pi 3 Model B.
A continuación se muestra una imagen con los elementos principales.



GPIO Raspberry Pi 3



Arduino nano

El Arduino Nano es una pequeña y completa placa basada en el ATmega328 (Arduino Nano 3.0) o el ATmega168 en sus versiones anteriores (Arduino Nano 2.x) que se usa conectándola a una protoboard. Tiene más o menos la misma funcionalidad que el Arduino Duemilanove, pero con una presentación diferente. No posee conector para alimentación externa, y funciona con un cable USB Mini-B.

Características:

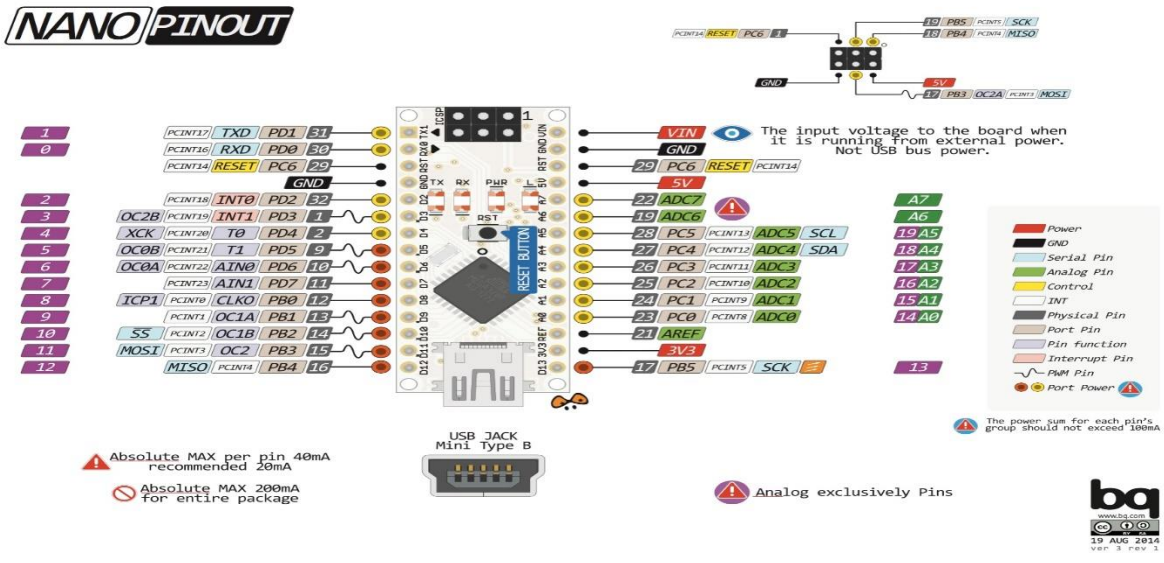
- Microcontrolador: Atmel ATmega328 (ATmega168 versiones anteriores)
- Tensión de Operación (nivel lógico): 5 V
- Tensión de Entrada (recomendado): 7-12 V
- Tensión de Entrada (límites): 6-20 V
- Pines E/S Digitales: 14 (de los cuales 6 proveen de salida PWM
- Entradas Analógicas: 8 Corriente máx por cada PIN de E/S: 40 mA
- Memoria Flash: 32 KB (ATmega328) de los cuales 2KB son usados por el bootloader (16 KB – ATmega168)
- SRAM: 2 KB (ATmega328) (1 KB ATmega168)
- EEPROM: 1 KB (ATmega328) (512 bytes – ATmega168)
- Frecuencia de reloj: 16 MHz
- Dimensiones: 18,5mm x 43,2mm

Energía

El Arduino Nano posee selección automática de la fuente de alimentación y puede ser alimentado a través de:

- Una conexión Mini-B USB.
- Una fuente de alimentación no regulada de 6-20V (pin 30).
- Una fuente de alimentación regulada de 5V (pin 27)

Diagrama de Pines:



Sensor Ultrasónico HC-SR04

El HC-SR04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el HC-SR04 se destaca por su bajo consumo, gran precisión y bajo precio por lo que está reemplazando a los sensores polaroid en los robots más recientes.

Características:

- Dimensiones del circuito: 43 x 20 x 17 mm
- Tensión de alimentación: 5 Vcc
- Frecuencia de trabajo: 40 KHz
- Rango máximo: 4.5 m
- Rango mínimo: 1.7 cm
- Duración mínima del pulso de disparo (nivel TTL): 10 µS.
- Duración del pulso eco de salida (nivel TTL): 100-25000 µS.
- Tiempo mínimo de espera entre una medida y el inicio de otra 20 mS.

Pines de conexión:

- VCC
- Trig (Disparo del ultrasonido)
- Echo (Recepción del ultrasonido)
- GND

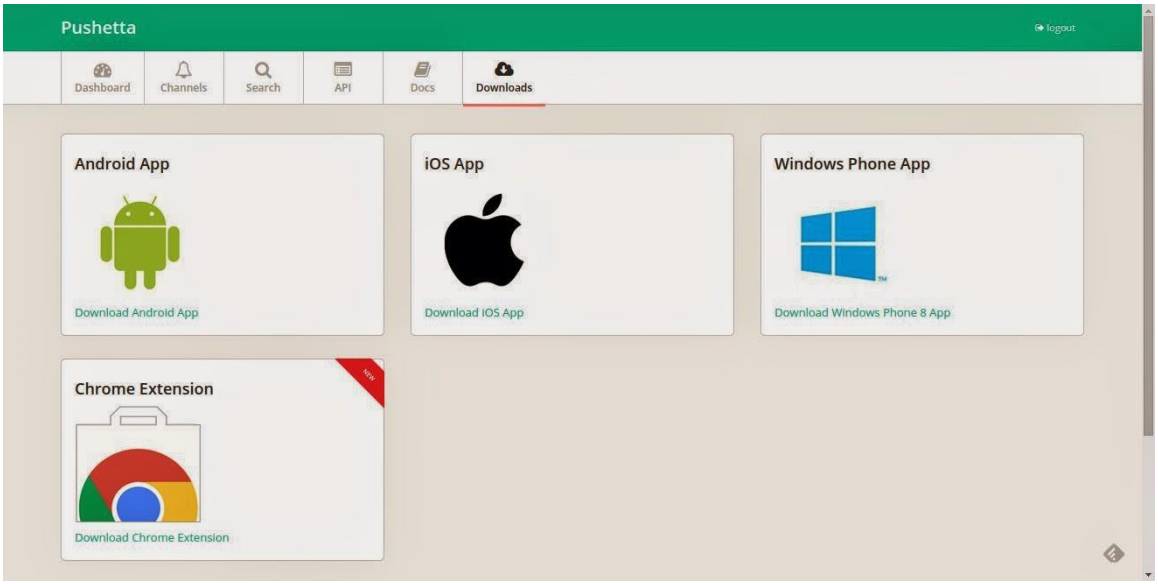
Distancia = {(Tiempo entre Trig y el Echo) * (V.Sonido 340 m/s)}/2

Pushetta

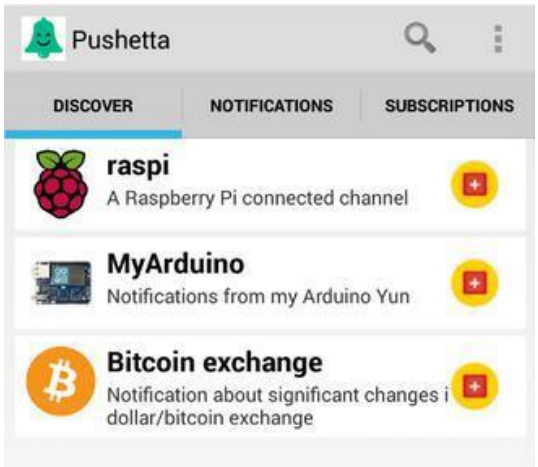
Es un servicio que permite la creación de canales y el envío de notificaciones a nuestro dispositivo móvil.

Las notificaciones push son un tipo de comunicación entre dispositivos donde es el servidor quien inicia la petición hacia el cliente cuando tiene una nueva notificación que enviar, permitiendo así un importante ahorro de recursos respecto a otras tecnologías más convencionales.

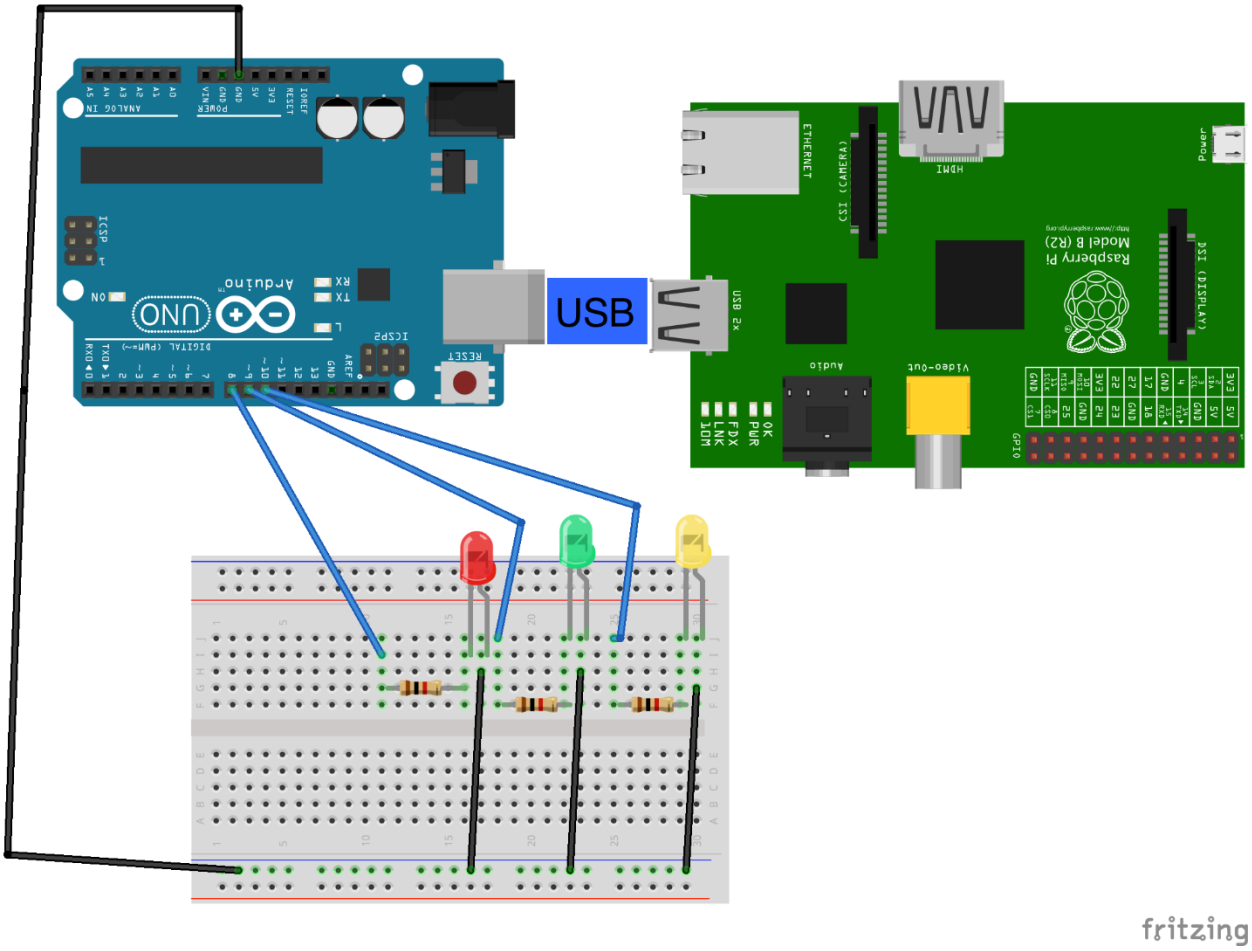
Pushetta permite crear de manera gratuita canales para enviar notificaciones a varios dispositivos (Android, iOS, Windows Phone y Chrome)



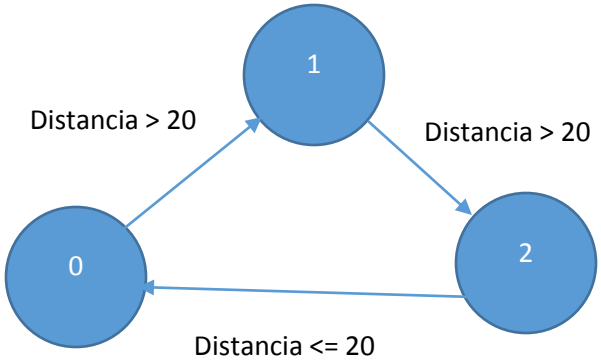
Pushetta trabaja por suscripciones, así que debes buscar el canal del cual quieres recibir las notificaciones y subscribirse. Si deseas recibir la notificación en tu dispositivo móvil es necesario descargar la aplicación de Pushetta y subscribirse al canal.



Diagrama



Máquina de Estados



Estado	Actual	Avisar
0	1	0
1	0	0
2	0	1

Librerías

Para realizar este proyecto se utilizaron las siguientes librerías para Raspberry:

PySerial:

Entre las muchas librerías de Python se encuentra una específica que provee la capacidad de utilizar muy fácilmente el o los puertos seriales de una computadora para comunicarse con otros dispositivos. Esta biblioteca se denomina pySerial y fue creada por Chris Liechti. Al igual que Python, pySerial es compatible con muchas plataformas o Sistemas Operativos como Linux, Windows, MacOS, BSD, Jython y IronPython

Time:

El módulo time de la biblioteca estándar de Python proporciona un conjunto de funciones para trabajar con fechas y/o horas.

Una fecha y/o hora específica se puede expresar de tres modos diferentes:

- como un número (float),
- como una cadena
- y como un objeto struct_time (es una tupla a medida (namedtuple) con nueve valores enteros).

Push:

Librería creada para enviar notificaciones push de la Raspberry a un dispositivo móvil que tenga instalada la aplicación de Pushetta para recibir la notificación.

urllib2:

Python nos facilita el módulo urllib2 para la interacción con URLs, tambien existe urllib. Pero aunque su uso es muy parecido, es menos completa que urllib2.
El módulo urllib2 puede leer datos de una URL usando varios protocolos como HTTP, HTTPS, FTP, o Gopher.

Json:

Esta librería principalmente parsea el JSON de archivos o strings. También parsea el JSON en un diccionario o en una lista en Python y viceversa, es decir, convierte los diccionarios y las listas de Python en cadenas JSON.

Código

Raspberry pi 3

Despachador.py

```
import
serial
        import time
        import push
        arduino = serial.Serial('/dev/ttyUSB0', 9600) #ttyUSB0 para arduino nano , ttyACMo
        para arduino uno
        arduino.open
        txt = ''
        print("........Bienvenido.......\n")
        print("Configuracion:\n")
        times = int(raw_input('Cuantas veces al dia desea alimentar a su perrito? '))
        #cuantas veces al dia
        horarios = []  #lista para agregar horarios
        while (times > 0):
              hora = raw_input('Horario formato 24h (HH:MM): ') #Input
              horarios.append(hora)
              times = times - 1
        porcion = int(raw_input("Tamanio de la porcion? 1=Chica, 2=Mediana, 3=Grande "))
        #porcion

        print("........Configuracion completa.......\n")
        print("........Horarios: \n")
        for i in horarios:
              print "horario: %s" % i
        print("........Porcion: \n")
        if porcion == 1:
              print("Porcion: Chica \n")
        if porcion == 2:
              print("Porcion: Mediana \n")
        if porcion == 3:
              print("Porcion: Grande \n")
        while True:
              hora_sys = time.strftime("%H:%M")
              for horarios_comp in horarios:     #Compara los horarios con la hora actual
                    if hora_sys == horarios_comp:
                          arduino.write(str(porcion)) #Mandar un comando hacia Arduino
```

```
                        time.sleep(60)

        #arduino.write("L") #solo para debug
        time.sleep(0.1)
        print hora_sys
        while arduino.inWaiting() > 0:
              txt = arduino.read(1)
              if txt == "9":
                    push.sendNotification("8969e7c6b9d30c765ab2dea18fbed44a244a880d",
  "pet-feeder", "Hola! Tu perrito tiene poca comida :( ")
              if txt == "8":
                    push.sendNotification("8969e7c6b9d30c765ab2dea18fbed44a244a880d",
  "pet-feeder", "Hola! Tu perrito ya tiene comida :) ")
              print txt
              txt = ''
  arduino.close() #Finalizamos la comunicación
```

Push.py

```
import urllib2
import json
#Esta es la funcion que usa el script para enviar las notificaciones
def sendNotification(token, channel, message):
        data = {
               "body" : message,
               "message_type" : "text/plain"
        }
        req =
urllib2.Request('http://api.pushetta.com/api/pushes/{0}/'.format(channel))
        req.add_header('Content-Type', 'application/json')
        req.add_header('Authorization', 'Token {0}'.format(token))
        response = urllib2.urlopen(req, json.dumps(data))
```

Arduino

Despachador.ino

```
long
distancia;
              long tiempo;
              int estado=0; //bandera que indica estado
              int actual=1; //bandera que indica si hay comida
              int avisar=0;  //bandera que indica si envio senial de comida
              void setup(){
              Serial.begin(9600);
              pinMode(9, OUTPUT);     //Declaramos el pin 9 como salida del pulso ultrasonico
              trig
              pinMode(8, INPUT);      //Declaramos el pin 8 como entrasa (tiempo que tarda en
              volver) echo
              pinMode(4, OUTPUT);     //motor1
              }
              void loop(){
                if (Serial.available()) { //Si está disponible el serial
                  char c = Serial.read(); //Guardamos la lectura en una variable char
                  if (c == '1') { //Si recibe 1 significa que sera una porcion Chica
                    digitalWrite (4, HIGH); //apaga motor
                    delay(2000);
                    digitalWrite (4, LOW); //apaga motor
                  } else if (c == '2') { //Si es 2 significa que sera una porcion Mediana
                    digitalWrite (4, HIGH); //apaga motor
                    delay(4000);
                    digitalWrite (4, LOW); //apaga motor
                  } else if (c == '3') { //Si es 3 significa que sera una porcion Grande
                    digitalWrite (4, HIGH); //apaga motor
                    delay(6000);
                    digitalWrite (4, LOW); //apaga motor
                  } else{
                    digitalWrite(4, LOW); //se apaga el led
                  }
                }


                  /* Se estabiliza el sensor */
                  digitalWrite(9,LOW);
                  delayMicroseconds(5);
                  /* Se envia el pulso ultrasonico */
```

```
    digitalWrite(9, HIGH);
    delayMicroseconds(10);
    /* Mide el tiempo transcurrido entre la salida y la llegada del pulso
ultrasonico */
    tiempo=pulseIn(8, HIGH);
    /* Se calcula la distancia on esta formila*/
    distancia= int(0.017*tiempo);
    delay(200);

  if((actual == 1) && (avisar == 0)){estado = 0;}
  if((actual == 0) && (avisar == 0)){estado = 1;}
  if((actual == 0) && (avisar == 1)){estado = 2;}

  switch(estado){
    case 0 :
      if(distancia > 20){ //hay poca comida
        actual=0;
        avisar=0;
      }else{
        actual=1;
        avisar=0;
      }
      break;

    case 1 :
          if(distancia > 20){
            actual=0;
            avisar=1;
            Serial.print("9");   //mandar senial a la raspberry para indicar que
hay poca comida
          }else{
            actual=1;
            avisar=0;
          }
          break;
    case 2 :
          if(distancia <= 20){
            actual=1;
            avisar=0;
            Serial.print("8");   //mandar senial a la raspberry 8= ya hay comida
          }else{
            actual=0;
            avisar=1; } break; }}
```

Conclusión y Resultados

El mundo de Internet está evolucionando rápidamente bajo la presión de una nueva ola, que está cambiando la forma de construir servicios de Internet y volviendo más fuerte el concepto de IoT ya que con el uso de la tecnología que en la actualidad contamos podemos brindarles a nuestras mascotas una mejor calidad de vida y contribuir con los dueños a realizar sus actividades de una manera más sencilla.

Al llevar a cabo este Proyecto, se pretende atender las necesidades básicas para el cuidado de los perros y facilitar al dueño la realización de estas actividades como: la alimentación de la mascota en este caso perros.

Referencias

Pushetta:
http://www.pushetta.com/